Version

2.3

SOUTH EASTERN EUROPE

NestPay®

Merchant Integration
3D Pay Hosting

www.asseco-see.com.tr

Document Information

Project/Product Name			
Project Manager			
Document Version No			
Document Code			
Prepared By		**Preparation Date**	22/01/2015
Reviewed By		**Review Date**	22/01/2015

Distribution List

From	Date	Phone/Fax

To	Action*	Due Date	Phone/Fax

* Action Types: Approve, Review, Inform, File, Action Required, Attend Meeting, Other (please specify)

Version History

Ver. No.	Ver. Date	Revised By	Description
1.5	1/22/2015	Okan GÜRBÜZ	Document format has been changed
1.6	1/2/2013	Nildem DEMİR	English parameters
1.7	1/29/2013	Nildem DEMİR	Recurring
1.8	4/12/2013	Selcuk Yılmaz	Explanation has been added about instalment and rnd.
1.9	7/15/2013	Nildem Demir	Address details
2.0	7/30/2013	Nildem Demir	Address details optional
2.1	4/10/2013	Nihal Müstecaplıoğlu	“comments” parameter added.
2.2	20/12/2013	Yiğit İPÇİOĞLU	An information about instalment is added.
2.3	14/01/2014	Nildem Demir	Callback added.

Proprietary Notice

Contents

1.3D Pay Hosting Model

3D Pay Hosting model is the basic internet integration model with payment page hosting, supporting 3D transactions.

Basic Properties:

- Enables processing of 3D secure card transactions
- HTTP Post method is supported for merchant integration
- Payment is done automatically by NestPay.

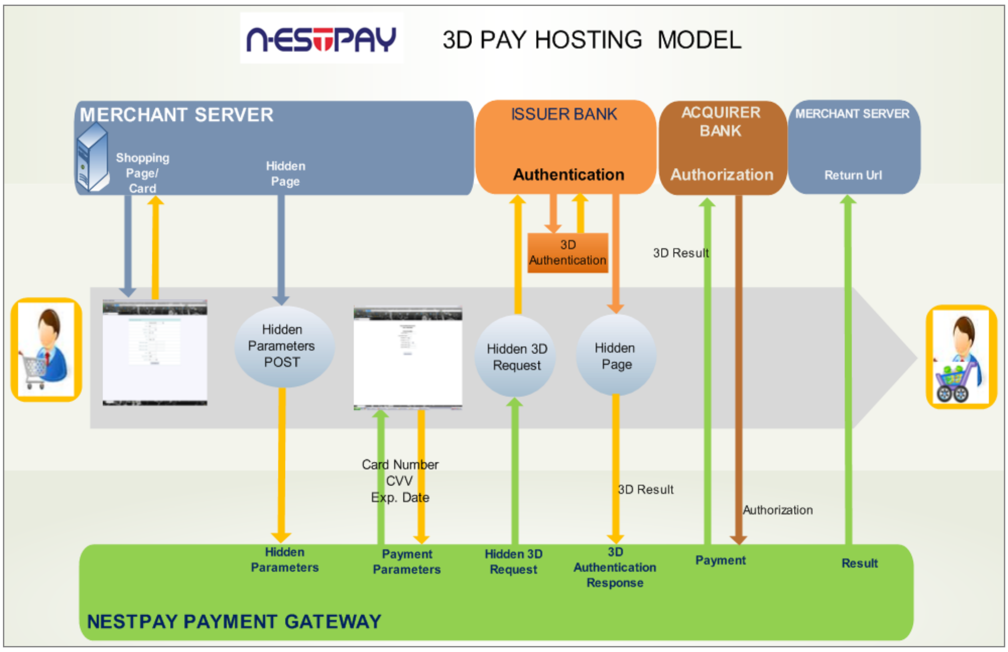
After obtaining all necessary shopping data from the customer like order amount, currency, customer name/surname etc., merchant server generates a unique order ID. Necessary parameters are posted using HTTP Post method to NestPay gateway.

For card payment methods (Visa, MasterCard etc.) merchant server needs to submit the card details like card number, CVV2, and expiry date information. After the order/card data is obtained from the user, 3D flow (enrolment and authentication queries) starts. In 3D flow, the 3D authentication information of the customer is queried by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods include using 3D secure passwords, one-time passwords, and security questions.

Using this model,

1. The customer knows that his/her personal information is not saved by the merchant, because credit card information is collected by NestPay, not the merchant.
2. Integration process is easy.
3. Bank's SSL certificate is used. Therefore the software cannot be updated.
4. In addition to the obligatory parameters, the merchant can POST its own data, such as username, user email or user id. This data is sent back to the merchant by the bank.

2. Nestpay 3D Pay Hosting Model



3DPayHosting Model Diagram

3. Quick Start Guide

This section will describe how to perform a successful Sale VISA transaction with **3D Pay Hosting Model**.

3.1 Generate Hash for Client Authentication

Hash is the base64-encoded version of the hashed text which is generated with SHA1 algorithm. To generate the hash for client authentication, following values should be appended with the given order:

plaintext = clientid + oid + amount + okurl + failurl + trantype + instalment + rnd + storekey ;

If instalment parameter is used, it must be included in hash function.

If instalment parameter is not used, then there is no need to include in hash function.

For instalments the system presumes that frequency is monthly.

rnd : It is a random parameter. Random string can be maximum 20 characters long.

For example, for the given parameters:

clientid	: 990000000000001
oid	: 1291899411421
amount	: 91.96
okurl	: https://www.teststore.com/success.php
failurl	: https://www.teststore.com/fail.php
trantype	: Auth
instalment	: 2
rnd	: asdf
storekey	: 123456

Hash would be:

plaintext = 990000000000001129189941142191.96
https://www.teststore.com/success.phphttps://www.teststore.com/fail.phpAuth**2**asdf**123456**
Hash = Base64(SHA1(plaintext))

3.2 Posting Hidden Parameters

Mandatory input parameters are posted to NestPay Payment Gateway located at **https://host/fim/est3dgate** as hidden parameters.

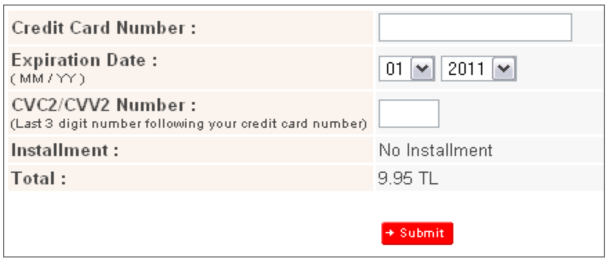
clientid	**:** Merchant ID (given by Nestpay)
storetype	**: "3d_pay_hosting"**
hash	**:** Hash value for client authentication
trantype	**:** "Auth"
amount	**:** transaction amount
currency	**:** ISO code of transaction currency (949 for TL)
oid	**:** Unique identifier of the order
okUrl	**:** The return URL to which **NestPay Payment Gateway** redirects the browser of the customer if transaction is completed successfully.
failUrl	**:** The return URL to which **NestPay Payment Gateway** redirects the browser of the customer if transaction is completed unsuccessfully.
lang	**:** Language of the payment pages hosted by NestPay ("tr" for Turkish, "en" for English)
encoding	**:** Page encoding

Sample HTTP form with mandatory parameter set

```
<form method="post" action="https://host/fim/est3dgate">
    <input type="hidden" name="clientid" value="990000000000001"/>
    <input type="hidden" name="storetype" value="3d_pay_hosting" />
    <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
    <input type="hidden" name="trantype" value="Auth" />
    <input type="hidden" name="amount" value="91.96" />
    <input type="hidden" name="currency" value="949" />
    <input type="hidden" name="oid" value="1291899411421" />
    <input type="hidden" name="okUrl" value="https://www.teststore.com/success.php"/>
    <input type="hidden" name="failUrl" value="https://www.teststore.com/fail.php" />
    <input type="hidden" name="lang" value="en" />
    <input type="hidden" name="rnd" value="asdf" />
    <input type="hidden" name="encoding" value="utf-8" />
</form>
```

3.3 VISA Payment Page

Consumer will enter his/her card details to complete the transaction and clicks the Pay button.



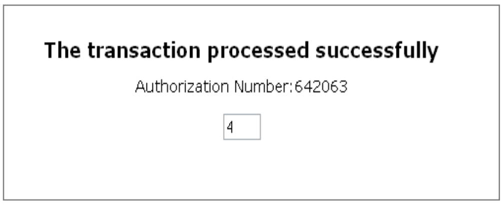
Credit Card Number :	
Expiration Date : (MM / YY)	01 2011
CVC2/CVV2 Number : (Last 3 digit number following your credit card number)	
Installment :	No Installment
Total :	9.95 TL

3.4 3D Authentication

In 3D flow, the 3D authentication information of the customer is collected by the issuer bank. The methods for 3D authentication can be different for different issuers. Examples of 3D authentication methods include using 3D secure passwords, one-time passwords, and security questions.

3.5 Transaction Result Page

The transaction result will be displayed to the customer. If the transaction is successful, the authorization code will be displayed. The customer will be redirected to *okUrl* if *refreshtime* has passed.



The transaction processed successfully

Authorization Number:642063

4

3.6 Merchant Success Page

If the transaction is successful, the customer will be redirected to **okUrl**, which is submitted on step 2 to NestPay Payment Gateway. All parameters posted by the merchant are returned back to the merchant. In addition to merchant parameters, gateway returns the transaction response parameters and MPI response parameters related to 3D secure transaction flow, which can be found in Appendix A.

Basic transaction response parameters for fully authenticated successful 3D transaction:

Response	: "Approved"
AuthCode	: Authorization code of the transaction
HostRefNum	: Host reference number
ProcReturnCode	: “00”
TransId	: Unique transaction ID
mdStatus	: “1”

For the example transaction above, the transaction response parameters would be:

Response	: "Approved"
AuthCode	**:** 544889
HostRefNum	**:** 034910000320
ProcReturnCode	**:** “00”
TransId	**:** 103491153310910033
mdStatus	**:** “1”

4.Integration Basics

4.1 HTTP Post Integration

After receiving a valid order, parameters are posted to NestPay payment gateway as hidden parameters with the HTTP form. In addition to mandatory parameters, the merchant can post order billing/shipping and order item details to payment gateway, which can be viewed later on Merchant Administration Panel. For optional parameters explanations please refer to Appendix A.

The 28 byte-long base-64 encoded xid parameter is the unique Internet transaction ID which is required for 3D secure transactions. If it is not sent by the merchant, it will be created automatically by the system.

4.1.1 Sample HTTP form with mandatory and optional parameters

```
<form method="post" action="https://host/fim/Nestpaygate">
    <input type="hidden" name="clientid" value="990000000000001"/>
    <input type="hidden" name="storetype" value="3d_pay_hosting" />
    <input type="hidden" name="hash" value="iej6cPOjDd4IKqXWQEznXWqLzLI=" />
    <input type="hidden" name="trantype" value="Auth" />
    <input type="hidden" name="amount" value="91.96" />
    <input type="hidden" name="currency" value="949" />
     <input type="hidden" name="instalment" value="">
    <input type="hidden" name="oid" value="1291899411421" />
    <input type="hidden" name="okUrl" value="https://www.teststore.com/success.php" />
    <input type="hidden" name="failUrl" value="https://www.teststore.com/fail.php" />
     <input type="hidden" name="callbackUrl" value="https://www.teststore.com/callback.php" />
    <input type="hidden" name="lang" value="tr" />
    <input type="hidden" name="rnd" value="asdf" />
    <input type="hidden" name="tel" value="012345678">
    <input type="hidden" name="email" value="test@test.com">
    <input type="hidden" name="encoding" value="utf-8" />

<!-- Billing Parameters [All Optional]-->
        <input type="hidden" name="BillToCompany" value="Billing Company">
        <input type="hidden" name="BillToName" value="Bill John Doe">
        <input type="hidden" name="BillToStreet1" value="Address line 1">
        <input type="hidden" name="BillToStreet2" value="Address line 2">
        <input type="hidden" name="BillToStreet3" value="Address line 3">
        <input type="hidden" name="BillToCity" value="Istanbul">
        <input type="hidden" name="BillToStateProv" value="mystate">
        <input type="hidden" name="BillToPostalCode" value="12345">
        <input type="hidden" name="BillToCountry" value="616">

<!-- Shipping Parameters [All Optional]-->
        <input type="hidden" name="printBillTo" value="true">
        <input type="hidden" name="ShipToCompany" value="Shipping Company">
        <input type="hidden" name="ShipToName" value="Ship John Doe">
        <input type="hidden" name="ShipToStreet1" value="Address line 1">
        <input type="hidden" name="ShipToStreet2" value="Address line 2">
        <input type="hidden" name="ShipToStreet3" value="Address line 3">
        <input type="hidden" name="ShipToCity" value="Istanbul">
        <input type="hidden" name="ShipToStateProv" value="mystate">
        <input type="hidden" name="ShipToPostalCode" value="12345">
        <input type="hidden" name="ShipToCountry" value="616">

<!-- Order Item Parameters [All Optional]-->
        <input type="hidden" name="printShipTo" value="true">
       <input type="hidden" name="ItemNumber1" value="a5">
       <input type="hidden" name="ProductCode1" value="a5">
       <input type="hidden" name="Qty1" value="3">
       <input type="hidden" name="Desc1" value="a5 desc">
       <input type="hidden" name="Id1" value="a5">
       <input type="hidden" name="Price1" value="6.25">
       <input type="hidden" name="Total1" value="7.50">

<!—Recurring Payment [All Optional]-->
       <input type="hidden" name="RecurringPaymentNumber" value="6">
       <input type="hidden" name="RecurringFrequencyUnit" value="M">
       <input type="hidden" name="RecurringFrequency" value="1">

</form>
```

4.2 Card Transactions

Submitting the form with card data will start 3D authentication flow with the customer. After the 3D authentication process is completed, MPI response parameters and all parameters sent by merchant will be post back to the merchant to make the payment. The payment will be done according to **mdStatus** field which shows the status code of the 3D secure transaction.

4.2.1 MPI Response Parameters

mdStatus : Status code for the 3D transaction

txstatus	3D status for archival
eci	Electronic Commerce Indicator
cavv	Cardholder Authentication Verification Value, determined by ACS.
md	Hash replacing card number
mdErrorMsg	Error Message from MPI

4.2.1.1 Possible mdStatus Values

- 1 = Authenticated transaction (Full 3D)
- 2, 3, 4 = Card not participating or attempt (Half 3D)
- 5, 6, 7, 8 = Authentication not available or system error
- 0 = Authentication failed

4.2.1.2 Successful Transaction

The authorization code will be displayed. The customer will be redirected to **okUrl** of the merchant server if *refreshtime* has passed. All input parameters along with transaction response parameters will be posted to **okUrl**, and the Response parameter will be "**Approved**"

4.2.1.3 Failed Transaction

The failure message will be displayed. The customer will be redirected to **failUrl** of the merchant server if refreshtime has passed. All input parameters along with transaction response parameters will be posted to **failUrl**, and the Response parameter will be "**Declined**" or "**Error**".

4.2.1.4 Transaction Response Parameters

Response	"Approved", "Declined" or "Error"
AuthCode	Authorization code of the transaction
HostRefNum	Host reference number
ProcReturnCode	Transaction status code
TransId	Unique transaction ID
ErrMsg	Error text (if *Response* "Declined" or "Error")
ClientIp	IP address of the customer
ReturnOid	Returned order ID, must be same as input oid
MaskedPan	Masked credit card number
PaymentMethod	Payment method of the transaction
rnd	Random string, will be used for hash comparison
HASHPARAMS	Contains the field names used for hash calculation. Field names are appended with ":" character
HASHPARAMSVAL	Contains the appended hash field values for hash calculation. Field values appended with the same order in *HASHPARAMS* field
HASH	Hash value of *HASHPARAMSVAL* and merchant password field

4.2.1.5 MPI Response Parameters

mdStatus	: Status code for the 3D transaction
txstatus	: 3D status for archival
eci	: Electronic Commerce Indicator
cavv	: Cardholder Authentication Verification Value, determined by ACS.
mdErrorMsg	: Error Message from MPI (if any)
xid	: Unique Internet transaction ID

4.2.1.6 Possible Transaction Results

- **Response:** "Approved"

ProcReturnCode will be "00". This shows that the transaction has been authorized.

- **Response:** "Declined"

ProcReturnCode will be a 2 digit number other than "00" and "99" which corresponds to acquirer error code. This shows that the transaction has NOT been authorized by the acquirer. *ErrMsg* parameter will give the detailed description of the error. For detailed description of acquirer error codes for *ProcReturnCode,* refer to Appendix B.

- **Response:** "Error"

ProcReturnCode will be "99". This shows that the transaction has NOT reached the acquirer authorization step. *ErrMsg* parameter will give the detailed description of the error.

4.3 Hash Checking

After merchant receives the parameters, a hash check needs to be done at the merchant's server for validating the parameters. Hash checking ensures that the message is sent by NestPay.

4.3.1 Generating the plain text for hash

The parameters used for hash calculation are the following: *clientid, oid, AuthCode, ProcReturnCode, Response, rnd, md, eci, cavv, mdStatus.* Depending on the type of transaction, a subset of these parameters will be included in the hash generation:

- **Non 3D-secure card transactions**

  clientid, oid, AuthCode, ProcReturnCode, Response, rnd

- **3D secure card transactions**

  clientid, oid, AuthCode, ProcReturnCode, Response, mdStatusi eci, cavv ,md, rnd

All the values corresponding to these parameters are appended with the same order. The resulting string will be the same as HASHPARAMSVAL parameter values. The merchant password is appended as the final value to the end of this string. The resulting hash is the base64-encoded version of the hashed text which is generated with SHA1 algorithm. Under normal conditions generated hash text must be the same as the HASH parameter value posted by NestPay payment gateway. If not, merchant should contact NestPay support team.

Example: Non 3D card transactions

4.3.1.1 Assuming that the transaction response parameters

clientid, oid, AuthCode, ProcReturnCode, Response, rnd

HASHPARAMSVAL	: 990000000000001129189941142132165400Approvedasdf
HASHPARAMS	: clientid:oid:ProcReturnCode:Response:rnd:
HASH	: CVJssbkrhIzqZXVTwGobciDZI+A=

The merchant hash text will be generated with clientid, oid, ProcReturnCode, Response, rnd, and store key of the merchant as secret hash element. Assuming store key is 123456,

```
plain = 990000000000001129189941142132165400Approvedasdf123456
```

And the merchant hash is based64-encoded(SHA1(plain)). The result hash must be the same as the returning parameter HASH.

Not: Merchant has to check Hash parameter of HASHPARAMS & HASHPARAMSVAL & HASH return values.

4.4 Callback

To receive automatic notification about payment results, *callbackUrl* parameter is used. All payment result parameters are posted to the *callbackUrl* address automatically similar to *okUrl*/*failUrl*. This prevents information loss if the cardholder closes the browser window before redirection to the merchant. Callbacks will be sent periodically every 5 minutes until the merchant responds to the callback with an "Approved" message which means that the callback is acknowledged. Callbacks can be managed on Merchant Center.

Additionally, a timeout callback message will be sent to *callbackUrl* for session timeout cases.

5.Code Samples

The following procedure is for 3D Pay Model. Values have been inserted testing purposes. Merchants must define variables according to their needs. These code samples are given as a reference.

5.1 ASP Code Sample

5.2 .Net Code Sample

5.3 JSP Code Sample

5.4 PHP Code Sample

6.APPENDIX A: Gateway Parameters

6.1 Mandatory Input Parameters

Parameter	**Description**	**Format**
clientid	Merchant ID	Maximum 15 characters
storetype	Merchant payment model	Possible values: "pay_hosting", "3d_pay", "3d", "3d_pay_hosting"
trantype	Transaction type	Set to "Auth" for authorization, "PreAuth" for preauthorization
amount	amount transaction amount	Use "." or "," as decimal separator, do not use grouping character
currency	ISO code of transaction currency	ISO 4217 numeric currency code, 3 digits
oid	Unique identifier of the order	Maximum 64 characters
okUrl	The return URL to which NestPay redirects the customer if transaction is completed successfully.	Example: http://www.test.com/ok.php
failUrl	The return URL to which NestPay redirects the customer if transaction is completed unsuccessfully.	Example: http://www.test.com/fail.php

lang	Language of the payment pages hosted by NestPay	"tr" for Turkish, "en" for English
rnd	Random string, will be used for hash comparison	Fixed length, 20 characters
hash	Hash value for client authentication	

6.2 Optional Input Parameters

Parameter	**Description**	**Format**
refreshtime	Redirection counter value to okUrl or failUrl in seconds.	Number
callbackUrl	The URL to which NestPay makes callback	Example: http://www.test.com/callback.php
encoding	Encoding of the posted data. Default value is "utf-8" if not sent	Maximum 32 characters
description	Description sent to MPI	Maximum 255 characters
comments	Kept as "description" for the transaction	Maximum 255 characters
instalment	Instalment count PS: If it will be without instalment, then the instalment parameter's value must be null.	Number
GRACEPERIOD	Grace period; postpones the payment of given months	Number (months)
email	Customer's email address	Maximum 64 characters
tel	Customer phone	Maximum 32 characters
BillToCompany	BillTo company name	Maximum 255 characters
BillToName	BillTo name/surname	Maximum 255 characters
BillToStreet1	BillTo address line 1	Maximum 255 characters
BillToStreet2	BillTo address line 2	Maximum 255 characters
BillToCity	BillTo city	Maximum 64 characters
BillToStateProv	BillTo state/province	Maximum 32 characters
BillToPostalCode	BillTo postal code	Maximum 32 characters
BillToCountry	BillTo country code	Maximum 3 characters

ShipToCompany	ShipTo company	Maximum 255 characters
ShipToName	ShipTo name	Maximum 255 characters
ShipToStreet1	ShipTo address line 1	Maximum 255 characters
ShipToStreet2	ShipTo address line 2	Maximum 255 characters
ShipToCity	ShipTo city	Maximum 64 characters
ShipToStateProv	ShipTo state/province	Maximum 32 characters
ShipToPostalCode	ShipTo postal code	Maximum 32 characters
ShipToCountry	ShipTo country code	Maximum 3 characters
idl	Id of item #l, required for item #l	Maximum 128 characters
itemnumberl	Item number of item #l	Maximum 128 characters
productcodel	Product code of item #l	Maximum 64 characters
qtyl	Quantity of item #l	Maximum 32 characters
descl	Description of item #l	Maximum 128 characters
pricel	Price of item #l	Maximum 32 characters
total1	Subtotal of item #l	Maximum 32 characters
RecurringPaymentNumber	Total number of payments for recurring payment	Number
RecurringFrequencyUnit	Frequency unit for recurring payment	1 char: D=Day,W=Week,M=Month, Y=Year
RecurringFrequency	Frequency of recurring payment	Number
printBillTo	Print BillTo address fields on payment page	"true" or "false". If not sent, billTo address details will not be printed
printShipTo	Print ShipTo address fields on payment page	"true" or "false". If not sent, shipTo address details will not be printed

6.3 Transaction Response Parameters

Parameter	**Description**	**Format**
AuthCode	Transaction Verification/Approval/Authorization code	6 characters
xid	Internet transaction identifier	28 characters
Response	Payment status	Possible values: "Approved", "Error", "Declined"
HostRefNum	Host reference number	12 characters

ProcReturnCode	Transaction status code	2 digits, "00" for authorized transactions, "99" for Nestpay errors, others for ISO-8583 error codes
TransId	Nestpay Transaction Id	Maximum 64 characters
ErrMsg	Error message	Maximum 255 characters
ClientIp	IP address of the customer	Maximum 15 characters formatted as "###.###.###.###"
ReturnOid	Returned order ID, must be the same as input orderId	Maximum 64 characters
MaskedPan	Masked credit card number	12 characters, XXXXXX***XXX
EXTRA.TRXDATE	Transaction Date	17 characters, formatted as "yyyyMMdd HH:mm:ss"
rnd	Random string, will be used for hash comparison	Fixed length, 20 characters
HASHPARAMS	Contains the field names used for hash calculation. Field names are appended with ":" character	Possible values "clientid:oid:AuthCode:ProcReturnCode:Response:rnd:" for non-3D transactions, "clientId:oid:AuthCode:ProcReturnCode:Response:mdStatus:cavv:eci:md:rnd:" for 3D transactions
HASHPARAMSVAL	Contains the appended field values for hash calculation. Field values appended with the same order in HASHPARAMS field	Fixed length, 28 characters
HASH	Hash value of HASHPARAMSVAL and merchant password field	Fixed length, 20 characters

6.4 MPI Response Parameters

Parameter	**Description**	**Format**
mdStatus	Status code for the 3D transaction	1=authenticated transaction 2, 3, 4 = Card not participating or attempt

		5,6,7,8 = Authentication not available or system error 0 = Authentication failed
merchantID	MPI merchant ID	15 characters
txstatus	3D status for archival	Possible values "A", "N", "Y"
iReqCode	Code provided by ACS indicating data that is formatted correctly, but which invalidates the request. This element is included when business processing cannot be performed for some reason.	2 digits, numeric
iReqDetail	May identify the specific data elements that caused the Invalid Request Code (so never supplied if Invalid Request Code is omitted).	
vendorCode	Error message describing *iReqDetail* error.	
PAResSyntaxOK	If PARes validation is syntactically correct, the value is true. Otherwise value is false.	"Y" or "N"
ParesVerified	If signature validation of the return message is successful, the value is true. If PARes message is not received or signature validation fails, the value is false.	"Y" or "N"
eci	Electronic Commerce Indicator	2 digits, empty for non-3D transactions
cavv	Cardholder Authentication Verification Value, determined by ACS.	28 characters, contains a 20 byte value that has been Base64 encoded, giving a 28 byte result.
xid	Unique internet transaction ID	28 characters, base64 encoded
cavvAlgorthm	CAVV algorithm	Possible values "0", "1", "2", "3"
md	MPI data replacing card number	Alpha-numeric
Version	MPI version information	3 characters l(ike "2.0")
sID	Schema ID	"1" for Visa, "2" for Mastercard
MdErrorMsg	Error Message from MPI (if any)	Maximum 512 characters